Diffusion coefficient calculation of single molecule tracking data with smt package[1]

Sheng Liu, Sun Jay Yoo
Carl Wu Lab, JHU

A basic task in single molecule tracking is to determine diffusion coefficient from molecule trajectories identified after initial image data acquisition. Trajectory of a molecule is presented as a table of x,y and z coordinates in the unit of pixel, which then can be converted to other measurement such as μm according to the resolution of the camera. The package SMT provides methods to calculate diffusion coefficient by using mean square displacement (MSD) based, displacement cumulative distribution function (CDF)-based, as well as hidden Markov model (HMM) based methods from such input file. This vignette explains the usage of the package.

Contents

1. Input data preparation

1.1 Storing trajectory information

The input file for SMT is the txt output file from Diatrack software (diatrack.org). Trajectory[2] is extracted from the txt output file and then stored in a two level list, 1) the first level is a list of folder names; and 2) second level is a list of tracks.

[1] This tutorial is based on smt v0.3, tut on version 0.4 will be available in next release.
[2] The term “trajectory” and “track” are used interchangeably in this text.
this tut is based on smt v0.3, tut on version 0.4 will come soon.

```
trackll=list(FOLDER=list(track=track))
```

A naming scheme is used in the source code of smt package as follow, “trackl” denotes a list of tracks; and “trackll”, denotes a list of folders, each containing the tracks. The coordinates of trajectory is stored in a table like data structure called data.frame.

```
# construct trackll from data.frame
trackl=list(track_1=dataframe_1,
            track_2=dataframe_2,
            track_3=dataframe_3,
            ...
            track_n=dataframe_n)

trackll=list(FOLDER_1=trackl_1,
             FOLDER_2=trackl_2,
             FOLDER_3=trackl_3,
             ...
             FOLDER_n=trackl_n,)
```

1.2 Read in Diatrack tracking data with readDiatrack() function.

```
library(smt)
# specify Diatrack output file path
folder=system.file("extdata","SWR1",package="smt")
# read in Diatrack output file, merge them into a single master list
trackll=readDiatrack(folder=folder, merge=T)
# see the structure of the list
str(trackll,max.level=2)
```

```
List of 1
 $ SWR1:List of 346
  ..$ mage6.1.4.1.1        :'data.frame':     4 obs. of  3 variables:
  ..$ mage6.2.2.2.2        :'data.frame':     2 obs. of  3 variables:
```

```
  ..$ mage6.4.9.3.3        :'data.frame':     9 obs. of  3 variables:
  ..$ mage6.5.8.4.4        :'data.frame':     8 obs. of  3 variables:
  ..$ mage6.11.12.5.5      :'data.frame':     12 obs. of  3 variables:
  ..$ mage6.12.9.6.6       :'data.frame':     9 obs. of  3 variables:
```

This folder named “SWR1” has in total 346 tracks. Folder name is after the first $ sign, track name is after the secondary $ sign. To see coordinates of an individual track, you can specify the folder name and the track name.

```
# specify the folder name and the track name
trackll[["SWR1"]]["mage6.1.4.1.1"]
# alternatively, specify the index of the folder and the track
trackll[[1]][1]
```

```
$mage6.1.4.1.1
      x     y z
1 40.32 49.36 1
2 41.03 47.33 1
3 41.05 48.43 1
4 40.90 49.09 1
```

The track name has the structure,

```
fileID.        frameID.       duration.      indexPerFile.   indexPerTrackll
```

From the name of first trajectory, we can see that,
the last 5 character of the Diatrack output file is “mage6” (fileID);
it starts show up in the first frame of the movie (frameID);
it last for 4 frames (duration);
it is the first track of this Diatrack output file (indexPerFile);
it is also the first track of this combined trackll list object (indexPerTrackll);

You can see more details of parameters for reading in tracks and specify structure of the trackll by typing “?readDiatrack” in the R console.

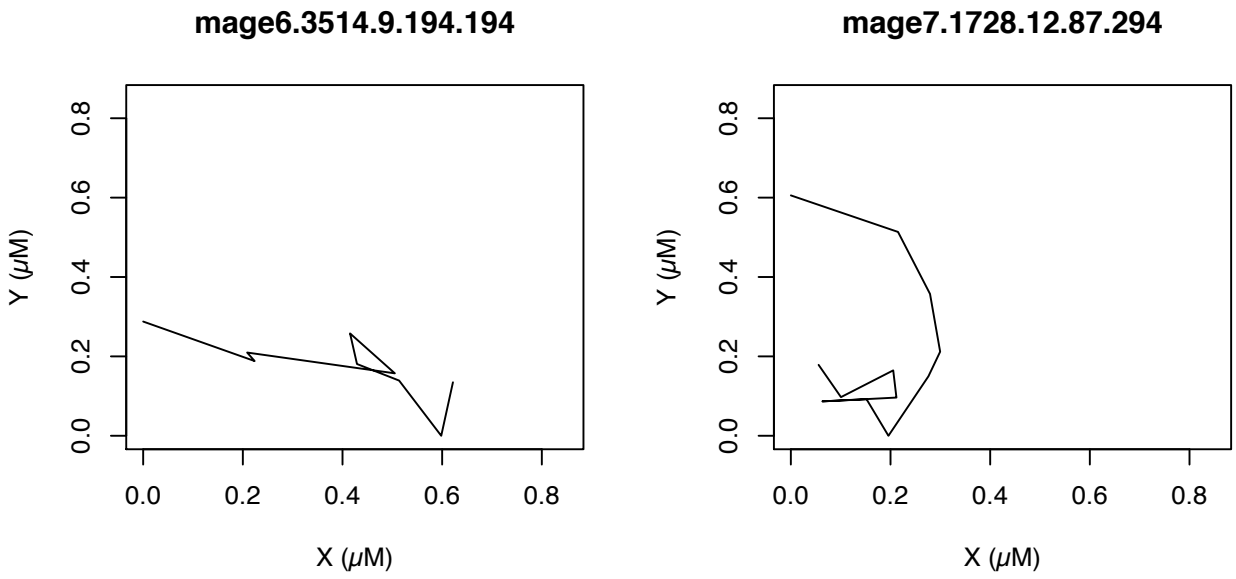
1.3 Plot tracks with plotTracks() function

The first thing one may want to do is to see how is the trajectory look like. The track name is useful in this case if one wants to see specific trajectories and its associated movies.

All one needs to do is to create a csv file contains trajectory names in its first column. smt package contains such an example csv file.

```
# specify the path of the file containing trajectory index names, index file
index.file2=system.file("extdata","INDEX","indexFile2.csv",package="smt")
# specify the folders containing the output files
folder1=system.file("extdata","SWR1",package="smt")
folder2=system.file("extdata","HTZ1",package="smt")
# plot trajectories specified in the trajectory index file
plotTrackFromIndex(index.file=index.file2,movie.folder = c(folder1,folder2))
```

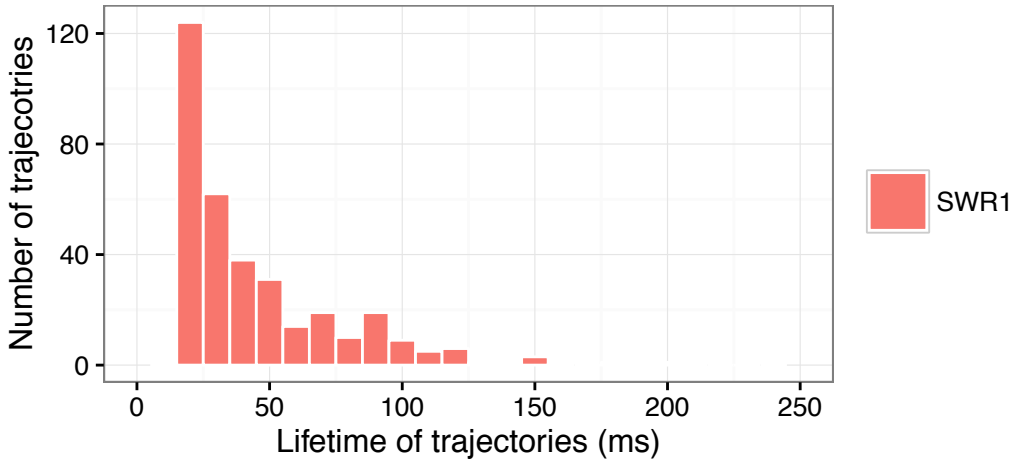
This plots the trajectories based on its trajectory name, with the information contained in its name (i.e. frameID and duration), one can also pull out its movie. See ?plotTrack for more plotting options.

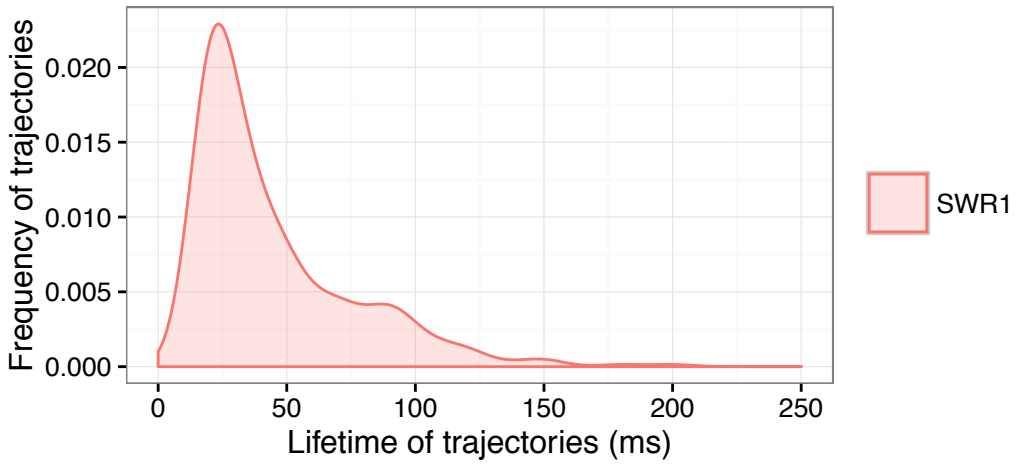


1.4 See distribution of the length of the trajectories with dwellTime() function.

One maybe interested to see the distribution of the length of the trajectories. This can be done by,

```
dwellTime(trackll,plot=TRUE) # default t.interval=10, x.scale=c(0,250)
```





We can see majority of the trajectory length centered at 20ms which is 2 frames. If we intend to only analyze frames length at specific range, one can use the filtration() function.

1.5 Filer trajectories on length with filtration() function

```
# select trajectories that have frame number greater than 7
trackll=filtration(trackll,filter=c(min=7,max=Inf))
```

To be continued…

2. Calculate diffusion coefficient using MSD based method

Mean square displacement
To see the displacement

2.1 Displacement profiling of individual trajectories using msd() function

2.2 Averaged of all trajectories
Mean square displacement